Name

Q1 - Décrire précisément ce qui se passe à chaque ligne.

```ruby
first_name = ["Pierre", "Paul", "Jacques"].sample
last_name = "Paillard"
puts "Great! #{first_name} #{last_name} is born"
```

1.
2.
3.

Q2 - Qu'est-ce qu'un constructeur? À quoi sert-il et comment l'appelle-t-on?

Q3 - Soit le code suivant. Quelle est la différence entre `string` et `String`?

```ruby
string = String.new("Hello World")
```

Q4 - Que fait la méthode suivante ?

```ruby
def mystery_method(pets)
  mammal_pets = []
  for pet in pets
    mammal_pets << pet if pet.mammal?
  end
  return mammal_pets
end
```

Q5 - Réimplémentez la méthode `mystery_method` en utilisant une des méthodes de `Enumerable`.

Q6 - Dans la question 4, quel peut être le type de `pets` ? Écrivez une classe `Animal` tel qu'un tableau d'instances de `Animal` passé à `mystery_method` renvoie un tableau vide.

Q7 - Écrivez une classe `Dog` qui hérite d'`Animal`, et qui définit la méthode d'instance adéquate vis-à-vis de la question 4.

Q8 - Soit la classe suivante. Comment appelle-t-on la "famille" de la méthode `overview` dans la classe ci-dessous ? Même question avec `catch_fish`.

```
class Fish
  def self.overview()
    return "Fishes can live in the sea"
  end
  def catch_fish
    # Does catch stuff
  end
end
```

Q9 - Écrivez du code qui appelle la méthode `overview` définie à la question précédente. Même question avec `catch_fish`

Q10 - Définissons deux classes, `Liquid` et `Alcohol`.

```ruby
class Liquid
  def drink
    "You drank some liquid"
  end
  def evaporate
    "There is no more liquid"
  end
end

class Alcohol < Liquid
  def drink
    "Yo are now drunk"
  end
end
```

Qu'est-ce qui est affiché par les deux dernières lignes du script suivant ?

```ruby
beer = Alcohol.new
beer.drink
beer.evaporate
```

2.
3.

Q11 - À quoi sert `super` ?

Q12 - Qu'est-ce qu'une variable d'instance ? Donner un exemple.

Q13 - À quoi sert `private` ? Quand l'utilise-t-on ? Donner un exemple.